MMIX in Class — Learning a Lot About Hardware Through Software

Axel Böttcher, Martin Ruckert and Matthias Utesch
Munich University of Applied Sciences
Department of Computer Science and Mathematics
{ab, ruckert, utesch}@cs.hm.edu
Overview – key message

• Learning about hardware through software providing continuity in the use of one processor simulation platform for various hardware related aspects of Computer Science

• Learning objectives according to Bloom’s Taxonomy: drive students towards conceptual, procedural and metacognitive knowledge dimensions

• The virtual MMIXprocessor: an appropriate programming and/or visualisation tool for a set of courses
The branding of our teaching is: science and application
Portfolio of hardware related courses
1. IT-Systems 1: structures of microprocessors at the register and register-transfer level
2. IT-Systems 2: hardware-/software interface up to basic operating system concepts
3. Computer Architecture: design and evaluation of modern multicore and superscalar processors
4. Compilers: designing back ends using the example of MMIX
MMIX – RISC-Computer for the third Millennium

- Existing processor simulators and visualisation tools often lack the ability to support multiple courses of a Computer Science curriculum
- In 1999 we decided to use MMIX as the right technology for our teaching:
  - State of the art technology
  - Education oriented towards international standards: „The Art of Computer Programming“
  - Simple to learn
  - Visualization: A community has developed tools
# Bloom´s Taxonomy (revised 2001)

<table>
<thead>
<tr>
<th>Knowledge Dimension</th>
<th>The Cognitive Process Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factual Knowledge</td>
<td>Remember</td>
</tr>
<tr>
<td>Conceptual Knowledge</td>
<td>Learning question:</td>
</tr>
<tr>
<td></td>
<td>Instruction question:</td>
</tr>
<tr>
<td>Procedural Knowledge</td>
<td>Assessment question:</td>
</tr>
<tr>
<td>Meta-cognitive Knowledge</td>
<td>Alignment question:</td>
</tr>
</tbody>
</table>

05.11.2009, © A.Böttcher, M.Ruckert, M.Utesch
<table>
<thead>
<tr>
<th>Learning Objectives</th>
<th>Use of MMIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Know basic components of a state-of-the-art processor</td>
<td>Knuth’s basic MMIX-simulator: Students can run and debug their programs</td>
</tr>
<tr>
<td>Implement control structures</td>
<td></td>
</tr>
<tr>
<td>Implement simple data structures</td>
<td></td>
</tr>
<tr>
<td>Program devices and simple device drivers</td>
<td>Knuth’s simulator tied to a virtual motherboard</td>
</tr>
</tbody>
</table>
# Computer Architecture

<table>
<thead>
<tr>
<th>Learning Objectives</th>
<th>Use of MMIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Know design of superscalar processors</td>
<td>Knuth’s pipeline simulator is used together with an Eclipse-based visualisation tool</td>
</tr>
<tr>
<td>Know and apply criteria of performance tuning and evaluation</td>
<td>Students can change configuration and observe the effects</td>
</tr>
</tbody>
</table>

05.11.2009, © A.Böttcher, M.Ruckert, M.Utesch
IT Systems courses

IT-Systems 1:
• Survey the Hardware/Software interface of IT-Systems from the register-transfer level to the operating system
• Write and test increasingly complex programs
→ MMIX is very well suited for this task because of its streamlined architecture and the good debugging support

IT-Systems 2:
• The Hardware/Software interface and the Application/Operating System interface are studied on a deeper level
→ Modified MMIX supports address translation, caching, programmable operating system calls, and asynchronous interrupts together with a virtual motherboard

05.11.2009, © A.Böttcher, M.Ruckert, M.Utesch
MMIX using a virtual motherboard

MMIX CPU

- I Cache 64 kByte
- D Cache 64 kByte

0x0000 – 0x1FFF

0x200000000 – 0x3FFFFFFFF

ROM 8kByte

RAM 4GByte

Virtual Bus

Virtual Motherboard

- Timer 32 Byte
- serial IO 16 Byte
- Frame Buffer 4 MByte
- Timer 16 Byte

05.11.2009, © A.Böttcher, M.Ruckert, M.Utesch
IT Systems course Evaluation

• Students use the familiar user interface and work with a complex and demanding system
• Students perceive the advanced simulator as easy to use as before (+0.08 on a scale from +2 to −2) and slightly more motivating to work with (+0.44 on the same scale)
• The consequent use of the same working environment, the same language, and the same target processor allows insights from various courses to complement each other, thereby reinforcing learned concepts and deepening the understanding of complex matters
The challenge - explain the behavior of modern superscalar processors:

- The ability of processors to simultaneously fetch several instructions and dispatch them to many parallel execution units
- Elaborated branch prediction schemes
- Multilevel caches

→ Meta MMIX simulator – extremely configurable
Meta MMIX simulator
Visualization – important tool in teaching

“Everything should, as far as possible, be placed before the senses. [...] If the objects themselves cannot be procured, representations of them may be used.”
[Johann Amos Comenius, 1657]

→ Modern results of neuroscience and neurodidactic confirm that especially for developing creativity it is important to exercise the right or visual and spatial side of the brain.

“Similarly [to researchers], educators can use visuals to engage students in active learning, support traditional lessons, and make their learning experience stronger and deeper.”
[McGrath and Brown, 2005]

→ Therefore, we built a visualisation environment on top of the complex pipeline simulation.
Screenshot of the mmmix visualisation

<table>
<thead>
<tr>
<th>The Fetch Buffer</th>
<th>The Pipeline Stages</th>
<th>The Reorder Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>BZ $22,@+8 (BZ)*</td>
<td>DIVI $23,$11,8 (div)*</td>
<td>LDO $20,$3,$13 (ld)*</td>
</tr>
<tr>
<td>ADDI $23,$23,1 (ADDI)</td>
<td>U2 GET $22,rR (get)*</td>
<td>BNZ $20,@+52 (br)*</td>
</tr>
<tr>
<td>SLUI $23,$23,3 (SLUI)</td>
<td>FPU1</td>
<td>SETL $22,1 (set)*</td>
</tr>
<tr>
<td>ADD $253,$253,$23 (ADD)</td>
<td>FPU2</td>
<td>STO $253,781,$13 (st)*</td>
</tr>
<tr>
<td>INCL $253,16 (INCL)</td>
<td>LSU1 STOI $22,64768,0 (st)*</td>
<td>SUBI $253,$253,8 (sub)*</td>
</tr>
<tr>
<td>JMP @+92 (JMP)</td>
<td>LSU2</td>
<td>STOI $22,64768,0 (st)*</td>
</tr>
<tr>
<td>SETL $22,0 (SETL)</td>
<td>CRU</td>
<td>DIVI $23,$11,8 (div)*</td>
</tr>
<tr>
<td>LDB $24,$253,$22 (LDB)</td>
<td>BRJ BNZ $20,@+52 (br)*</td>
<td>GET $22,rR (get)*</td>
</tr>
<tr>
<td>LDB $25,$20,$22 (LDB)</td>
<td>VEX1</td>
<td></td>
</tr>
<tr>
<td>CMP $23,$24,$25 (CMP)</td>
<td>VEX2</td>
<td></td>
</tr>
<tr>
<td>BNZ $23,@+16 (BNZ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BZ $24,@+52 (BZ)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

05.11.2009, © A.Böttcher, M.Ruckert, M.Utesch
Case Study – Activity View

The Activity view shows about 300 clock-cycles (300 ns at 1GHz)
Conclusion

“Whatever is taught should be taught as being of practical application in every-day life and of some definite use”
[Johann Amos Comenius 1657]

• Teaching hardware through software
• Bloom’s Taxonomy: define the learning path from remembering Factual Knowledge via applying Procedural Knowledge up to Metacognitive Knowledge
• The introduction of MMIX to our curriculum has proved to be an unconditional success
• The experiences with visualisation encourages us to exploit Serious Gaming for teaching: e.g. Software-Engineering, Computer Architecture, Economics